EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Mazour (reg. 59,318) on 2/11/2009.

2. The application has been amended as follows:

1. (Currently Amended) A method for developing an application, the method comprising:

defining file borders for development objects of an application in a data model represented using UML comprising a customizable extension, wherein the data model includes a component class, [[and]] a model class associated with the component class, and a controller class[[,]] that is associated with the component class, [[that]] wherein the controller class associates a user interface view to a business application model;

storing the development objects of the application in a file-based repository based on the file borders; and

employing an API derived from the data model to access the development objects,

wherein the API incorporates the customizable extension in the data model,

wherein during the API derivation, an endpoint of a relation between the data model classes is used to maintain the correct direction of the relation,

wherein to create methods for the API, a distinction between singular and plural forms of the endpoint of the relation between the data model classes is made to differentiate between interaction with one instance at the endpoint or with multiple instances at the endpoint,

wherein employing the API further comprises using tools that use the API to enable a user to perform a development operation; and

wherein the development operation comprises enabling [[a]] the user to refactor a copied one of the development objects to modify a key attribute of the copied one of the development objects while updating existing references within a scope of the refactor without invalidating all existing references.

6. (Currently Amended) The method of claim 1, further comprising enabling [[a]] the user to define a source path for one of the development objects.

10. (Currently Amended) The method of claim 1, further comprising enabling [[a]] the user to define the scope of the refactor.

12. (Currently Amended) A computer-implemented method for developing applications in a development system, the method comprising:

generating a data model for an application, the data model being implemented in a unified modeling language that includes a customizable extension, the data model including a feature defined using the customizable extension, wherein the data model includes a component class, [[and]] a model class associated with the component class, and a controller class[[,]] associated with the component class[[that]] wherein the controller class associates a user interface view to a business application model;

deriving an API from the data model, the API incorporating the feature, wherein during the API derivation, an endpoint of a relation between the data model classes is used to maintain the correct direction of the relation, and

wherein to create methods for the API, a distinction between singular and plural forms of the endpoint of the relation between the data model classes is made to differentiate between interaction with one instance at the endpoint or with multiple instances at the endpoint;

enforcing constraints specified in the data model by employing the derived API during development of the application; and

enabling a user to access development objects of the application using the API, wherein tools that use the API enable [[a]] the user to perform a development operation;

wherein the development operation comprises enabling [[a]] the user to refactor a copied one of the development objects to modify a key attribute of the copied one of the development objects while updating existing references within a scope of the refactor without invalidating all existing references.

19. (Currently Amended) A ~~computer-readable storage medium~~ machine-readable storage device storing a program for causing a computer to perform a method for developing an application, the method comprising:

defining file borders for development objects in a data model represented using UML comprising a customizable extension, wherein the data model includes a component class, [[and]] a model class associated with the component class, and a controller class[[,]] that is associated with the component class, [[that]] wherein the controller class associates a user interface view to a business application model;

caching the development objects from a repository storing the development objects using the file borders defined in the data model into a local development cache;

employing a user interface development tool that uses an API to access the development objects,

wherein the API is derived from the data model,

wherein during the API derivation, an endpoint of a relation between the data model classes is used to maintain the correct direction of the relation,

wherein to create methods for the API, a distinction between singular and plural forms of the endpoint of the relation between the data model classes is made to differentiate between interaction with one instance at the endpoint or with multiple instances at the endpoint, and

wherein using the API enables a user to perform a development operation; and

employing a second user interface development tool to enable [[a]] the user to refactor a copied one of the development objects to modify a key attribute of the copied

one of the development objects while updating existing references within a scope of the refactor without invalidating all existing references.

20. (Currently Amended) The ~~computer-readable storage medium~~ machine-readable storage device of claim 19, wherein the method further comprises storing the repository in a repository server.

21. (Currently Amended) The ~~computer-readable storage medium~~ machine-readable storage device of claim 19, wherein the user interface development tool comprises one of a project browser, an application modeler, a view designer, a controller and context editor, and a model editor.

23. (Currently Amended) The method of claim 12, further comprising enabling [[a]] the user to define a scope of the refactor using the tools, wherein the user selects a value for the scope.

Examiner's Statement of Reason(s) for Allowance

3. Claims 1-6, 8, and 10-23 (renumbered as 1-21) are allowed.

4. The following is an examiner's statement of reason s for allowance:

The closest prior arts of record, i.e. Template, Mitchell et al., Mortensen et al., Kemper, Schloegel et al., Hu, Stone, Barrett et al., Hrebejk et al., taken alone or in

combination, fail to teach or fairly suggest at least: during the API derivation, an endpoint of a relation between the data model classes is used to maintain the correct direction of the relation, wherein to create methods for the API, a distinction between singular and plural forms of the endpoint of the relation between the data model classes is made to differentiate between interaction with one instance at the endpoint or with multiple instances at the endpoint, wherein employing the API further comprises using tools that use the API to enable a user to perform a development operation as recited in the independent claims 1, 12, and 19.

While Template discloses building SNAP applications, Mitchell discloses creating named relations between classes in a dynamic OOP environment using mappers, Barrett discloses developing a flexible component-based software system by generating a UML model where the model is transformed to a meta model and a modifying the meta model via an API and a met modeler by a modeling tool, Mortensen discloses customization of metadata that describes objects in a software system without requiring modification of source code, Kemper discloses performing a refactoring operation to adjust a user's selection of source code, Schloegel discloses generating code for the model-based development of a system using graphical modeling entities, Hu discloses modeling XML applications using an extended UML notation, Stone discloses providing a UML diagram of a program for display in a GUI of development and a browser providing refactoring of the program, customizing the UML display and viewing documentation, and Hrebejk discloses creating a meta object facility including a model of MOF that is based upon a stored definition of MOF, ultimately, the cited references, taken alone or in combination, fail to teach or

fairly suggest at least: during the API derivation, an endpoint of a relation between the data model classes is used to maintain the correct direction of the relation, wherein to create methods for the API, a distinction between singular and plural forms of the endpoint of the relation between the data model classes is made to differentiate between interaction with one instance at the endpoint or with multiple instances at the endpoint, wherein employing the API further comprises using tools that use the API to enable a user to perform a development operation as recited in the independent claims 1, 12, and 19.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to INSUN KANG whose telephone number is (571)272-3724. The examiner can normally be reached on M-R 7:30-6 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis A. Bullock, Jr. can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status

information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Insun Kang/
Examiner, Art Unit 2193